JAKUB KISLINGER

CONNECTED INFORMATION FROM GIVEN ENTROPIES

SPOJENÁ INFORMACE ZE ZADANÝCH ENTROPIÍ

BACHELOR'S THESIS

Czech Technical University in Prague
Faculty of Electrical Engineering
Department of Cybernetics
Prague, May 2024
Supervisor: doc. Ing. Tomáš Kroupa, Ph.D.

Jakub Kislinger: *Connected Information from Given Entropies*, © May 2024.

E-MAIL:
kislijak@fel.cvut.cz

BACHELOR‘S THESIS ASSIGNMENT

I. Personal and study details

Student's name:	**Kislinger Jakub**	Personal ID number:	**507361**
Faculty / Institute:	**Faculty of Electrical Engineering**		
Department / Institute:	**Department of Cybernetics**		
Study program:	**Open Informatics**		
Specialisation:	**Artificial Intelligence and Computer Science**		

II. Bachelor’s thesis details

Bachelor’s thesis title in English:

Connected Information from Given Entropies

Bachelor’s thesis title in Czech:

Spojená informace ze zadaných entropií

Guidelines:

The maximization of entropy [3] subject to moment/marginal constraint is a standard optimization problem that benefits from a variety of established solution approaches. Nevertheless, some variations of this entropy problem deviate from this framework due to non-convex constraints, rendering traditional numerical optimization techniques inapplicable. A pertinent example is the assessment of higher-order interactions in complex stochastic systems such as the human brain [1]. This assessment hinges on a concept known as connected information, requiring entropy maximization subject to entropic constraints on lower-dimensional marginals. Such constraints introduce non-convexity of the feasible set, making the problem challenging for conventional optimization methods. The objectives of this bachelor thesis are outlined as follows.
1. To consider the entropy problem within the context of entropic constraints, particularly focusing on the computation of connected information, and to evaluate the effectiveness of existing numerical methods in this setting using the implementation in Julia/JuMP [4].
2. To approximate the aforementioned problem through information theory methodology. This involves reinterpreting the original problem by introducing entropic variables and linear constraints that effectively capture the essence of entropy vectors [3].
3. To develop a linear programming-based solver using the Julia programming language. This solver will be designed for smaller-scale versions of the approximated problem from item 2. The study will also assess how well this method scales in terms of variable count and state space dimensions.
4. To test the accuracy of the approximation using real-world data derived from physical system measurements [1] and from the perspective of learning undersampled probability distributions [2].

Bibliography / sources:

[1] Elliot A. Martin, Jaroslav Hlinka, and Jörn Davidsen. "Pairwise network information and nonlinear correlations." Physical Review E 94.4 (2016): 040301.
[2] Ilya Nemenman, Fariel Shafee, and William Bialek. "Entropy and inference, revisited." Advances in neural information processing systems 14 (2001).
[3] Raymond W. Yeung. Information Theory and Network Coding, Springer (2008).
[4] Miles Lubin, et al. "JuMP 1.0: recent improvements to a modeling language for mathematical optimization." Mathematical Programming Computation (2023): 1-9

Name and workplace of bachelor’s thesis supervisor:

doc. Ing. Tomáš Kroupa, Ph.D. Artificial Intelligence Center FEE

Name and workplace of second bachelor’s thesis supervisor or consultant:

Date of bachelor’s thesis assignment: **21.01.2024** Deadline for bachelor thesis submission: **24.05.2024**

Assignment valid until: **21.09.2025**

doc. Ing. Tomáš Kroupa, Ph.D. Supervisor’s signature	prof. Dr. Ing. Jan Kybic Head of department’s signature	prof. Mgr. Petr Páta, Ph.D. Dean’s signature

III. Assignment receipt

The student acknowledges that the bachelor’s thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor’s thesis, the author must state the names of consultants and include a list of references.

. Date of assignment receipt	Student’s signature

AUTHOR STATEMENT FOR UNDERGRADUATE THESIS

I hereby declare that I have completed the presented thesis on my own and that I have included all the used sources of information in the list of references, in accordance with the *Methodological Guideline on Ethical Principles in the Preparation of University Theses*.

Prague, May 2024

. . . .
Jakub Kislinger

ACKNOWLEDGEMENTS

I would like to thank doc. Ing. Tomáš Kroupa, Ph.D. , to whom I am grateful for his invaluable guidance and support during this bachelor's thesis. His expertise and encouragement have been instrumental in its development.

ABSTRACT

Information entropy is a measure of the uncertainty of a probability distribution. Stochastic systems tend to be in the state of maximal entropy. Without any internal dependencies, this would correspond to a uniform distribution. However, variables are typically dependent on each other. We can gain insights into groups of mutually dependent variables by fixing marginal entropies and then computing the maximal possible entropy. When comparing the results, we can say how much groups of n variables interact compared to groups of other sizes. This thesis focuses on methods that maximise entropy under entropic constraints, the implementation and comparison of these methods, and their demonstration through several examples. It also evaluates data from real-life neurological experiments, comparing the results with data published by the authors and with data from another bachelor's thesis.

ABSTRAKT

Informační entropie je míra nejistoty pravděpodobnostní distribuce. Stochastické systémy mají tendenci být ve stavu maximální entropie. Bez jakýchkoliv vnitřních závislostí by to distribuce odpovídala rovnoměrnému rozložení. Ale většinou jsou jednotlivé proměnné na sobě závislé. Fixováním marginálních entropií a poté spočítáním maximální možné entropie jsme schopni získat náhled do skupin vzájemně závislých proměnných. Při pohledu na výsledky pak můžeme říci, jak moc všechny skupiny n proměnných spolu interagují v porovnání se skupinami jiných velikostí. Tato práce se zaměřuje na metody maximalizace entropie s entropickými podmínkami, jejich implementaci a porovnání a předvedení na několika příkladech. Také vyhodnocuje data z reálného neurologického experimentu a porovnává výsledky s originálními výsledky publikovanými autory a s výsledky jiné bakalářské práce.

KEYWORDS

Entropy, Entropy Maximisation, Entropic Constraints, Connected Information

KLÍČOVÁ SLOVA

Entropie, Maximalizace entropie, Entropické podmínky, Spojená informace

CONTENTS

LIST OF TABLES

1 INTRODUCTION

Probability distributions are hidden in many aspects of everyday life, and they provide information about phenomena that might seem to be completely random. Those stochastic systems usually require more than one variable to be described. When a probability distribution depends on more variables, it also contains information about the dependence of variables on each other. While mutual dependence of specific variables can be easily seen by eliminating other variables, insight into mutual dependence over every possible set of variables with given size cannot be seen that straightforwardly.

Inspection of those relations takes advantage of the fact that stochastic systems tend to be in the state of maximal entropy. So, the entropy of distribution is being maximised while preservation of specific properties helps with getting the insight. One of those properties is marginal distribution, which leads to a standard optimisation problem with established solutions.

However, fixing marginal distribution is applicable only for smaller systems with a suitable size of a probability distribution. For larger instances, different properties of probability distribution have to be fixed. One suitable property on which this thesis will focus is the entropy of marginal distributions.

The main goal of this thesis will be the implementation of methods computing the maximal information entropy while fixing the entropy of marginal distributions of given orders. Those methods will then be tested on real-world data and from the perspective of their usage on undersampled distributions. In the end, the importance of connected information will be clarified through several experiments.

1.1 PROPOSED SOLUTIONS

Maximisation of information entropy with fixed marginal entropies is a challenging optimisation task mainly due to the non-convex nature of information entropy. This thesis will implement and evaluate two forms of the maximisation task

- placing constraints on the marginal entropies and submitting the non-convex task to solvers,
- and reformulating and approximating the task by introducing entropic variables and linear constraints.

After implementing methods for estimating the maximal entropy of a distribution, the estimation of entropy from real data samples will be focused. Entropy will be estimated from the data

- by creating a probability distribution using the empirical distribution,
- and using NSB algorithm [13], specifically made for undersampled distributions.

These methods of estimation will be compared based on the number of data samples.

1.2 THESIS STRUCTURE

This chapter introduces the overall topic and outlines the proposed solutions. In Chapter 2, concepts from information theory will be presented along with the concepts required for the reformulated and relaxed task. Chapter 3 describes the data and tools used for the evaluation while the comparison of used methods is presented along with the experiments using real-world data in Chapter 5. The implementation of all methods is outlined in Chapter 4. Finally, Chapter 6 concludes the thesis with achieved goals and ideas for future research.

2 INFORMATION THEORY

This chapter introduces the concept of *entropy*, its significance, and the maximisation problem. In Information theory, the unit of information is the bit, representing two values. Therefore, it is conventional to use the binary logarithm to express results in the correct units. In this paper, we use $\log = \log_2$.

2.1 NOTATION

We introduce a notation to deal with interactions across multi-dimensional spaces. Let $N = \{1, \ldots, n\}$ for some positive n. Consider a non-empty finite set $\chi_i \subset \mathbb{R}$ for all $i \in N$; for $J \subset N$ define $\chi_J = \times_{i \in J} \chi_i$ and $\chi_N = \chi$. We will be dealing with a discrete random vector $\mathbf{X} = (X_1, \ldots, X_n)$ with the sample space χ and a probability mass function p, which satisfies the following conditions:

$$p : \chi \to [0, 1] \qquad \sum_{\mathbf{x} \in \chi} p(\mathbf{x}) = 1$$

When dealing with sets of indices, we often use notation without commas and parentheses; for example, $ijk = \{i, j, k\}$.

2.2 INFORMATION ENTROPY

As Shannon [18] proved, *information entropy* is a unique measure of the information provided by a probability distribution over a set of variables. It also quantifies the average uncertainty of a random variable.

Definition 1. The *entropy* $H(\mathbf{X})$ of a discrete random variable $\mathbf{X}$, which takes values from the sample space χ with a probability mass function p, is defined as

$$H(\mathbf{X}) = -\sum_{\mathbf{x} \in \chi} p(\mathbf{x}) \log p(\mathbf{x}) \tag{1}$$

If the probability $p(\mathbf{x})$ is equal to 0, then $p(\mathbf{x}) \log p(\mathbf{x})$ is also 0.

Information entropy can be interpreted as the number of bits needed to describe the probability distribution, as illustrated by Examples 1 and 2.

Example 1. *Consider a uniform distribution of a two-dimensional binary random vector $\mathbf{X}$ with a probability mass function*

$$p(0, 0) = p(0, 1) = p(1, 0) = p(1, 1) = 0.25.$$

The best way to encode this distribution is to let the first bit represent the first value and the second bit represent the second value. There is always the same probability for 0 *and for* 1 *(marginal distributions* p_1 *and* p_2*).*
On average, we need 2 *bits to describe the output, which corresponds to the entropy:*

$$H(\boldsymbol{X}) = 4 \cdot (-0.25 \log 0.25) = 2.$$

Example 2. *Consider a distribution with two binary values and a probability mass function, as shown in Table 1.*

Huffman coding [7] optimally reduces uncertainty by half with each bit of information and creates prefix-free coding. The first bit should indicate whether the sample is $(0,0)$ *or any other value (both options have a probability of 0.5). Let* 0 *correspond to* $(0,0)$*. If* 0 *is received, the sample is known exactly. If* 1 *is received, we need to distinguish among three samples, and the updated probabilities are now the following:*

$p(0,0)$	0.5
$p(0,1)$	0.25
$p(1,0)$	0.125
$p(1,1)$	0.125

Table 1: The probability mass function used in Example 2.

$$p(0,1) = 0.5; \quad p(1,0) = 0.25; \quad p(1,1) = 0.25.$$

The second bit will decide if the sample is $(0,1)$ *(encoded as* 10*) or a different value (encoded as* $11\_$*), again reducing the uncertainty by half. If the second bit is* 1*, the third bit will determine whether the sample is* $(1,0)$ *(encoded as* 110*) or* $(1,1)$ *(encoded as* 111*).*

sample	coded as
$(0,0)$	0
$(0,1)$	10
$(1,0)$	110
$(1,1)$	111

Table 2: An example of an ideal coding for the samples from Example 2 with probabilities according to Table 1.

The average number of bits, n_{avg}*, needed to encode a sample using Huffman coding [7] can be calculated from Tables 1 and 2:*

$$n_{avg} = 0.5 \cdot 1 + 0.25 \cdot 2 + 0.125 \cdot 3 + 0.125 \cdot 3 = 1.75$$

, which also corresponds to the entropy

$$H(\boldsymbol{X}) = -0.5 \log 0.5 - 0.25 \log 0.25 - 2 \cdot 0.125 \log 0.125 = 1.75$$

2.3 MARGINAL DISTRIBUTIONS

Definition 2. Let $N = \{1, ..., n\}$, non-empty subset $J \subseteq N$, and $\chi_J = \times_{i \in J} \chi_i$. The *marginal distribution* p_J is a probability distribution given by

$$p_J(\mathbf{x}) = \sum_{\mathbf{y} \in \chi_{N \setminus J}} p(\mathbf{x}, \mathbf{y}), \quad \mathbf{x} \in \chi_J$$

The probability distribution p_J is called the *J-marginal of p* and has an order $|J| = k$. If $J = \{i\}$, we write p_i.

For a 3-dimensional probability distribution, a visualisation of marginal distributions is shown in Figure 1.

Definition 3. Consider χ_i to be a non-empty sample space. Let $N = \{1, ..., n\}$ and a non-empty subset $J \subseteq N$. Let $\chi_J = \times_{i \in J} \chi_i$, $\mathbf{X}_J = (X_i)_{i \in J}$, and p_J be the J-marginal of p. The *J-marginal entropy* $H(\mathbf{X}_J)$ is defined as:

$$H(\mathbf{X}_J) = -\sum_{\mathbf{x} \in \chi_J} p_J(\mathbf{x}) \log p_J(\mathbf{x})$$

The marginal entropies for a 3-dimensional probability distribution are also shown in Figure 1.

This thesis focuses on maximising the information entropy of a random vector $\mathbf{X}$, where the J-marginal entropy is consistent between the final and original distribution of $\mathbf{X}$ for all $J \subset N$ where the order of J is less than or equal to a given constant.

2.4 MAXIMISATION OF INFORMATION ENTROPY

The maximal value of information entropy without any constraints is achieved when the distribution is uniform. This can be demonstrated using Lagrange multipliers:

$$\begin{aligned}
L &= -\sum_{\mathbf{x} \in \chi} p(\mathbf{x}) \log p(\mathbf{x}) - \lambda \left(\sum_{\mathbf{x} \in \chi} p(\mathbf{x}) - 1 \right) \\
\frac{\partial L}{\partial p(\mathbf{x})} &= -\log p(\mathbf{x}) - 1 - \lambda \\
\frac{\partial L}{\partial \lambda} &= \sum_{\mathbf{x} \in \chi} p(\mathbf{x}) - 1 \\
\forall \mathbf{x} \in \chi : p(\mathbf{x}) &= 2^{-1-\lambda} \\
\forall \mathbf{x} \in \chi : p(\mathbf{x}) &= \frac{1}{|\chi|} \qquad (2)
\end{aligned}$$

Equation (2) shows that maximum entropy is achieved for a uniform distribution.

Our task is to maximise information entropy while the underlying distribution satisfies certain conditions. Specifically, we will fix the marginal information entropies for some of the marginal distributions. This problem does not have a general analytical solution, and therefore, we need to use

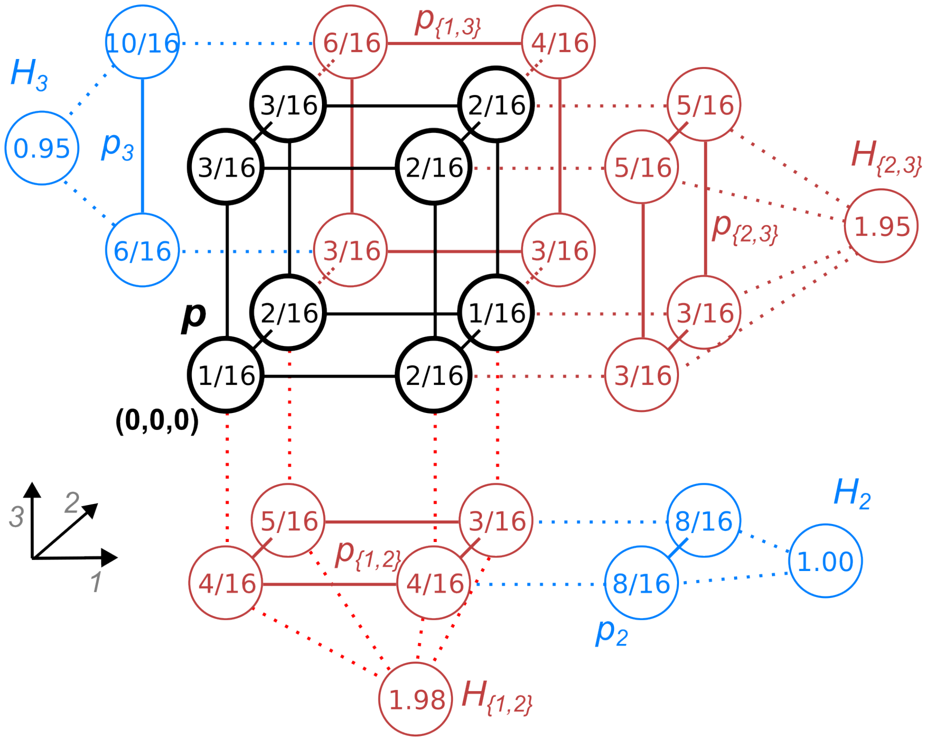

Figure 1: Visualised probability distribution (black) with the corresponding marginal distributions (p) of orders 2 (red) and 1 (blue) and the corresponding marginal entropies (H).

approximation methods to find the optimal solution. Moreover, this is not a convex optimisation problem, which will present itself as a problem during the optimisation.

Given an input distribution p, we aim to solve the following task:

$$\text{Maximise } H_q(\mathbf{X}) \text{ subject to } H_q(\mathbf{X}_J) = H_p(\mathbf{X}_J),\ \forall J : |J| \leqslant k, \tag{3}$$

where q is the vector of control variables that still satisfies the basic properties of a probability distribution.

Note that we do not need to recover the entire distribution q, but only its information entropy.

We will denote the optimal value of (3) as $H_p^k(\mathbf{X})$.

Using the chain rule for entropy and the fact that conditioning on a variable cannot increase the entropy,

$$H(X_1, \ldots, X_n) = H(X_1|X_2, \ldots, X_n) + \cdots + H(X_n) \leqslant \sum_{i=1}^{n} H(X_i), \tag{4}$$

where the equality holds if and only if the variables are independent. The resulting distribution is then the product of all marginals of size 1.

Since increasing the maximal order of the marginals only adds more conditions, the following inequalities hold:

$$\sum_{i \in N} H(\mathbf{X}_i) = H^1(\mathbf{X}) \geqslant H^2(\mathbf{X}) \geqslant \ldots \geqslant H^{n-1}(\mathbf{X}) \geqslant H^n(\mathbf{X}) = H(\mathbf{X}) \tag{5}$$

2.5 CONNECTED INFORMATION

An interesting characteristic of a probability distribution is the dependency of variables on each other. The measure that describes the dependence of the variables in all groups of a given size is called connected information [16].

Definition 4. Let p be a probability distribution of a random vector $\mathbf{X} = (X_1, \dots, X_n)$ on the sample space χ. The *connected information* of order k (where $k = 2, \dots, n$) is defined as:

$$I_p^k(\mathbf{X}) = H_p^{k-1}(\mathbf{X}) - H_p^k(\mathbf{X}). \tag{6}$$

$\boxed{\text{Without loss of generality, we will be using } I^k(\mathbf{X}) = I_p^k(\mathbf{X}).}$

Note that connected information is always non-negative due to the inequalities (5). The value $I^k(\mathbf{X})$ reflects the level of *k-order* stochastic interactions among the random variables $(X_1, \dots, X_n) = \mathbf{X}$ with the probability distribution p. If all variables are independent, then $I^k(\mathbf{X}) = 0\ \forall k:\ k = 2, \dots, n$. Examples are shown and computed in Chapter 5.

2.6 POLYMATROID CONE

Definition 5. Let $\mathcal{P}(N)$ be the power set of N. A *polymatroid* is a function $h : \mathcal{P}(N) \to \mathbb{R}$ that satisfies the following conditions:

1. $h(\emptyset) = 0$,
2. $h(A) \geqslant h(B)$ for all $A, B \subseteq N$ with $A \supset B$, *monotonicity*
3. $h(A) + h(B) \geqslant h(A \cup B) + h(A \cap B)$ for all $A, B \subset N$. *submodularity*

Definition 6. A *polymartoid cone (of order n)*, labelled Γ_n, is the set of all polymatroids on $\mathcal{P}(N)$, where $|N| = n$.

Note that for any $h_1, h_2 \in \Gamma_n$, any non-negative linear combination is also a polymatroid:

$$\alpha_1 h_1 + \alpha_2 h_2 \in \Gamma_n \quad \alpha_1, \alpha_2 \geqslant 0.$$

In other words, a polymatroid cone [4] is a convex cone in the space of all functions $\mathcal{P}(N) \to \mathbb{R}$.

To fully describe a polymatroid cone, it is sufficient to use $n + 2^{n-2}\binom{n}{2}$ *elemental* linear inequalities. Firstly, n equations for monotonicity,

$$\forall i \in N:\ h(N) \geqslant h(N \setminus i), \tag{7}$$

and secondly, $2^{n-2}\binom{n}{2}$ equations for submodularity,

$$\begin{aligned}&\forall i, j \in N, i \neq j, \forall A \subseteq N \setminus ij:\\ &\qquad h(A \cup i) + h(A \cup j) \geqslant h(A \cup ij) + h(A).\end{aligned} \tag{8}$$

Basic examples of polymatroids are entropic vectors, introduced in Section 2.7.

2.7 ENTROPIC REGION

Definition 7. Let p be a probability distribution of a random vector $\mathbf{X} = (X_1, \dots, X_n)$. An *entropic vector* is defined as a function h_p, which satisfies the following conditions:

$$h_p(J) = \begin{cases} H_p(\mathbf{X}_J) & J \neq \emptyset, J \subseteq N, \\ 0 & J = \emptyset. \end{cases}$$

Due to the basic properties of entropy, an entropic vector satisfies all the conditions required of a polymatroid, hence $h_p \in \Gamma_n$.

Definition 8. Let Γ_n^* be the set of all entropic vectors h_p derived from a distribution p of a random vector $\mathbf{X} = (X_1, \dots, X_n)$. Then Γ_n^* is called an *entropic region (of order n)*.

A *Shannon-type inequality* is an inequality that can be expressed as a non-negative linear combination of inequalities from (7) and (8). However, these inequalities are not sufficient to describe all entropic regions. Zhang and Yeung [21] demonstrated the following:

- $\Gamma_2^* = \Gamma_2$.
- $\Gamma_3^* \subset \Gamma_3$. However, for the topological closure, $\overline{\Gamma_3^*} = \Gamma_3$, and therefore, it is fully represented by the Shannon-type inequalities.
- $\overline{\Gamma_4^*} \subset \Gamma_4$. There exist non-Shannon type inequalities that constrain the topological closure and are valid for an entropic region, so-called *Zhang-Yeung inequalities*:

$$\begin{gathered} 3[h(ik) + h(il) + h(kl)] + h(jk) + h(jl) \\ -h(i) - 2[h(k) + h(l)] - h(ij) - 4h(ikl) - h(jkl) \geqslant 0 \\ \text{for all distinct } i, j, k, l \in N \end{gathered} \tag{9}$$

  These inequalities are valid for $n \geqslant 4$.

  Since k and l are interchangeable, there are $\dfrac{n!}{2 \cdot (n-4)!}$ unique Zhang-Yeung inequalities satisfied by every entropic vector.

3 DATASETS

Testing entropy maximisation methods requires datasets suitable for inspecting interactions among differently sized sets of variables. Using randomly generated samples without any modification is not a suitable approach since the random generator tries to return uniformly distributed values. Connected information of these distributions is minimal, meaning the methods would be tested with very atypical conditions.

The best way of obtaining data is to get samples from real-life systems. In our thesis, we will be using data from Martin et al. [11] (Section 3.1) and from Fokoue and Gunduz [6] (Section 3.2).

3.1 MAGNETIC RESONANCE IMAGING SIGNAL

Martin et al. [11] used in their paper dataset consisting of time series of functional magnetic resonance imaging signals. The samples originate from 96 healthy volunteers who were monitored in IKEM (Institute for Clinical and Experimental Medicine) in Prague. Data contain 36480 samples, each corresponding to one time-point. Each sample represents 20 signals from different regions of the brain, 10 from the default mode network and 10 from the fronto-parietal network.

Signals in the MAT file format are expressed as floating-point numbers. Since our methods expect discrete probability distributions, we have discretised the data. According to [11], the signals in their experiment were discretised into 2 and 3 levels using equiquantal (equiprobable) binning. Following the same principle, we partitioned the signals into the same levels and added levels 4 and 5.

After this process, the final samples were 10-dimensional vectors of discrete values from 1 to 10 for two brain regions. Those vectors were used during computations in Section 5.2.

Data from Martin et al. [11] are not publicly available, but they were provided upon request.

3.2 STUDENT QUESTIONNAIRE

Fokoue and Gunduz [6] conducted research among students at Gazi University in Ankara (Turkey). They were given a survey about a course and its instructor containing 28 questions. Questions were answered with integers ranging from 1 to 5, and the responses were saved along with five additional attributes.

The resulting dataset contains 5820 sets of student answers - a table of 33 columns (5 + 28) and 5820 rows. For computations, additional attributes were ignored due to inconsistency in value ranges with the rest of the questions. No additional changes to the dataset were needed.

For benchmarking in Section 5.1, columns $6 \dots 6 + (d - 1)$ were used based on the number of dimensions required d. In Section 3, specific columns from the table were chosen. Specifically, questions 13, 14, 16, 18, 20, 21, 22, and 28 were selected because they all reflect the teacher's subjective view. Due to this connection, interaction among different questions is expected.

Data, including the questions, can be obtained at `https://archive.ics.uci.edu/dataset/262/turkiye+student+evaluation`, 6.5.2024; data are in CSV format.

4 METHODS

The primary goal of this thesis is to implement methods to compute connected information. It involves finding the maximal information entropy while fixing the marginal entropies of specified orders. The Julia programming language was chosen for this purpose. The result is a Julia package providing an interface for methods to maximise the information entropy of a given distribution and other useful functions. This package is available for use under EntropyMaximisation[1]. Part of the package also focuses on the maximisation of entropy with fixed marginal distributions (rather than their entropies).

Julia was chosen mainly for its strengths in mathematical computing, including intuitive and efficient handling of multi-dimensional arrays, high efficiency (as a partially compiled language, it is faster than most other math/optimisation-focused languages like Python or R), and a convenient way of publishing the resulting package.

We used JuMP [10], a Julia modelling language for mathematical optimisation.

The resulting package offers various entropy maximisation methods, with the default method determined by the performance on real data, as demonstrated in Section 5.1.

4.1 DIRECT MAXIMISATION USING SOLVERS

Direct maximisation involved rewriting the maximisation task (3) into Julia, specifically the JuMP modelling language. The optimiser aimed to find the distribution with the maximum entropy while satisfying the entropic conditions.

Since the entropic constraints were in the form

$$\sum_{x \in \mathcal{X}_J} p_J(\mathbf{x}) \log p_J(\mathbf{x}) = \mathrm{const.},$$

Non-linear programming (NLP) solvers had to be used.

4.1.1 Maximisation Using the Exponential Cone

The initial idea was to reformulate the objective of entropy maximisation (3) into the form of the exponential cone [1]. This was the first-choice approach because the exponential cones had been proven to be the most efficient solution to the maximisation problem with fixed marginal distributions (not their entropies).

[1] https://gitlab.com/kislijak/entropy-maximisation

The entropy equation (1) was rewritten into the form of the exponential cone as follows:

$$K_{\exp} = \{(x,y,z) \in \mathbb{R}^3 : ye^{x/y} \leqslant z, y > 0\}, \tag{10}$$

which could be solved by Second Order Cone Programming (SOCP) solvers. Each part of the sum (1) was rewritten as follows:

$$\begin{aligned} t_i &\leqslant -p(i)\log(p(i)) \\ p(i)\log(p(i)) + t_i &\leqslant 0 \\ \log(p(i)) + \frac{t_i}{p(i)} &\leqslant 0 \\ p(i)e^{\frac{t_i}{p(i)}} &\leqslant 1 \end{aligned} \tag{11}$$

Now Equation (11) could be easily rewritten into the form of Equation (10) with the following substitutions:

$$x = t_i; \quad y = p(i); \quad z = 1 \tag{12}$$

To fix the marginal entropies, the entropies of the sums over all of the remaining dimensions were fixed. The sums then created the marginal distributions.

To allow the solver to maximise the entropy, it was provided with the corresponding variables and constraints (all variables and constraints were created by Julia macros):

1. Variable p represented the optimal probability distribution with maximised entropy. Its length was equal to the length of the original distribution. To ensure that p was a probability distribution, a constraint was introduced for the sum of p, which had to be equal to 1, and for the values, which had to be non-negative.

2. The fixed marginal entropies were ensured by placing equality between each entropy of the original marginal (the sum over the remaining dimensions of the original distribution) and the corresponding entropy of the model's marginal. Visualisation of marginal entropies connected to their corresponding marginals is shown in Figure 2. Since entropy (1) is not a linear function, non-linear constraints were used for the solver.

3. Another introduced variable was t, an array of lower bounds of partial products of the probability mass function and its logarithm:

$$t_i \leqslant -p(i)\log(p(i)). \tag{13}$$

4. The final constraints were placing the variables into the exponential cone. Using the already derived assignment (12), each point in the distribution was put into the constraints.

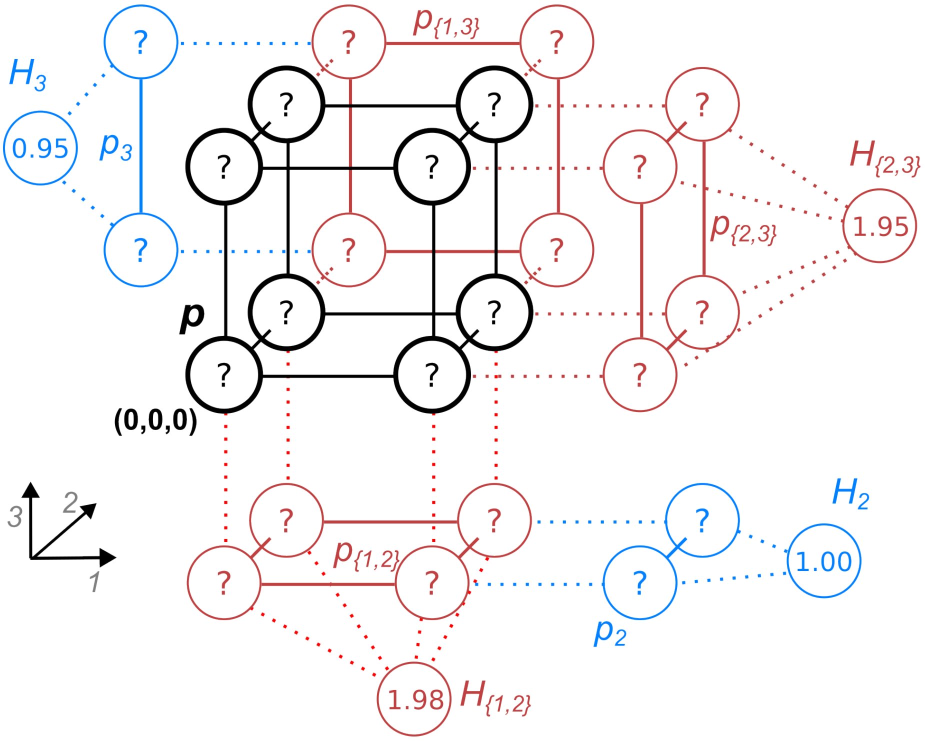

Figure 2: Visual representation of maximisation task with fixed marginal entropies (H) - probability distribution (black) and the corresponding marginal distributions of orders 2 (red) and 1 (blue) with unknown values. Some of the corresponding marginal entropies are shown with values according to Figure 1.

Since the distribution's entropy is the sum over partial products (1), our goal was to maximise the sum over t, while considering the introduced variables.

After setting all of the variables, constraints, and the main objective, the solver optimised the model. The optimised value was the sum over its objective values.

Note that the exponential cone used the logarithm of the base 10. Therefore, the last step was to divide the value by $\log_1 0(2)$ to return the information entropy, which uses the logarithm of base 2.

By maximising the sum over t using the exponential cone, the goal was reached at a point where the inequality in (13) became equality

$$t_i = -p(i)\log(p(i)).$$

4.1.2 Maximisation Using the Basic Entropy Equation

To directly maximise entropy with entropic constraints, the second approach involved putting the constraints in their original form (1) into the model. The main difference was that the constraint of the exponential cone was not used, which allowed the use of other solvers, primarily focused on NLP.

While the implementation of the entropic constraints remained the same as in maximisation using the exponential cone (Subsection 4.1.1), a significant difference was in the definition of objective variable t. In this method, t was defined as:

$$t_i = -p(i)\log(p(i)).$$

Similarly to the previous approach, the objective remained the same - maximisation of the sum of t - the entropy of the whole distribution.

4.2 RELAXATION OF THE MAXIMISATION TASK

Another approach to computing the maximal entropy of a distribution was to relax the constraints, making the task (3) easier for the solver to solve. The entropic region (Section 2.7) was used for this purpose.

Note that the entropic region can represent a possible entropy function of a distribution using only linear inequalities. So, when we maximised the entropy, we obtained the entropy as a function value of the entropic vector. However, this did not tell us what the underlying distribution looked like, only what the marginal entropies were while achieving the maximum overall entropy.

The following was the relaxed linear optimisation task:

$$\text{Maximise } h(N) \text{ subject to } h(J) = H(J) \text{ for all } J \subseteq N \text{ where } |J| \leqslant k, \tag{14}$$

where $H(J)$ is the J-marginal entropy of the original distribution.

The first step of the maximisation was to define the polymatroid (Section 2.6). The definition is shown in Algorithm 1.

Algorithm 1: Elemental constraints of polymatroid

```
Data: Function $h : \mathcal{P}(N) \to \mathbb{R}$
$h(\emptyset) = 0$
for $i$ in $N$ do
    $h(N) \geqslant h(N \setminus i)$
end
for $A$ in $powerset(N, max\_size = n - 2)$ do
    for $i, j$ in $powerset(N \setminus A, size = 2)$ do
        $h(A \cup i) + h(A \cup j) \geqslant h(A \cup ij) + h(A)$
    end
end
```

When Zhang-Yeung inequalities (9) were used in the maximisation, they had to be defined for the polymatroid as well (Algorithm 2).

The last step of setting the constraints was to fix the entropies. This was an easier task than fixing the entropies during the direct optimization (Section 4.1) because here the model variables were the entropies and not the values of the probability distribution. Entropic constraints were, therefore, linear (Algorithm 3).

Finally, the task (14) was optimized, and the resulting value was $h(N)$. Note that there were no conversions of the result since the entropy was already encoded in bits because entropy in the constraints was also encoded in bits.

Algorithm 2: Zhang-Yeung inequalities for polymatroid

```
Data: Function $h : \mathcal{P}(N) \to \mathbb{R}$
if $n \geqslant 4$ then
    for $i, j$ in $\mathtt{powerset}(N, \mathtt{size} = 2)$ do
        for $k, l$ in $\mathtt{powerset}(N \setminus ij, \mathtt{size} = 2, \mathtt{ordered} = \mathtt{true})$ do
            $0 \leqslant 3[h(ik) + h(il) + h(kl)] + h(jk) + h(jl) - h(i) - 2[h(k) + h(l)] - h(ij) - 4h(ikl) - h(jkl)$
        end
    end
end
```

Algorithm 3: Fixing entropy for all marginals

```
Data: Function $h : \mathcal{P}(N) \to \mathbb{R}$, joined probability p, maximal order of
      fixed marginal entropy m_size
for $i \leftarrow 1$ to $m\_size$ do
    $\mathtt{marginals} = \mathtt{permutations}(n, \mathtt{length} = i)$
    for $m$ in $\mathtt{marginals}$ do
        $p_m = \mathtt{marginal\_distribution}(p, m)$
        $h(m) = \mathtt{entropy}(p_m)$
    end
end
```

4.3 FULL ENTROPY REPRESENTATION OF DISTRIBUTION

In the previous section we had a method for determining the maximal entropy of the entire distribution, however, it was not possible to easily determine whether the Zhang-Yeung inequalities (9) restricted the resulting polymatroid. For this purpose, the Julia package Polyhedra [9] was used.

The Polyhedra package supports the same interface as JuMP, making it easy to replicate the optimisation algorithm from Section 4.2. No specific solver was defined; instead, the Polyhedra package internal solver returned the vertex representation of the polymatroid.

4.4 ESTIMATION OF ENTROPY FROM SAMPLES

So far, we have focused on maximising the entropy of a probability distribution. However, we have not taken into account that the data used to create the distribution could be undersampled, particularly the data from Martin et al. [11]. Data is considered undersampled when $N \ll K$, where N is the number of samples and K is the size of the sample space.

The following methods are compared in Subsections 5.1.2 and 5.1.3.

4.4.1 Estimation from Empirical Distribution

When there are sufficient samples, the most convenient method to estimate the entropy of the distribution would be using *empirical distribution* (frequency estimation). Given the lack of any further (prior) information, the distribution p over the sample space χ, $|\chi| = K$, was constructed as an empirical distribution based on N samples $x_1, \ldots, x_n$. The probability function of the empirical distribution was defined as:

$$p(x = i) = \frac{1}{N} \sum_{j=1}^{N} [x_j = i]$$

for all $i = 1, \ldots, K$, where $[x_j = i]$ evaluates to 1 only if $x_j = i$, otherwise it evaluates to 0. This distribution p served as the base for our computations.

Even for enough samples ($N \gg K$), the empirical distribution remains inaccurate. As demonstrated by Schürmann and Grassberger [17], the empirical distribution underestimates the entropy H_{emp}, which can be corrected with the following equation:

$$H = H_{emp} + \frac{K-1}{2N} + O\left(\frac{1}{N^2}\right) \tag{15}$$

However, this is a problem when the distribution is undersampled.

4.4.2 NSB Estimator

For undersampled datasets, Nemenman, Shafee, and Bialek [13] proposed a method for entropy estimation with insufficient samples called NSB. The method assumes a distribution with insufficient samples and no knowledge about the priors. An implementation exists in Matlab and Octave [12]. The implementation in Octave was used, and instead of creating the underlying distribution p and estimating its (marginal) entropies, the samples were categorized and formatted appropriately as inputs for the Octave code. The code was then called directly from Julia as a command in the command line.

4.5 SOLVERS

To compute the task, the methods outlined in Sections 4.1 and 4.2 rely on optimisation solvers written in JuMP modelling language [10] [2]. The tasks contain different constraints, and, therefore, the solvers vary for each method.

For the maximisation using the exponential cone (Subsection 4.1.1), the solver has to comprehend both non-linear constraints and the exponential cone constraints. The only solver supporting Non-Linear Programming (NLP) and Second Order Cone Programming covering (SOCP), which covers the exponential cone constraints, is currently Pajarito [5].

[2] https://jump.dev/JuMP.jl/stable/installation/#Supported-solvers, 2.5.2024, is a website with a list of all supported solvers by JuMP.

The maximisation method using only the basic entropy equation does not require the support of SOCP. The only requirement is the ability to calculate with non-linear constraints. For this purpose, publicly available solvers Ipopt [20] and MadNLP [19] were used.

Lastly, the optimisation using polymatroids requires only linear constraints. Linear constraints can be solved by most of the available solvers. The solvers used in Section **??** were SCS [14], a publicly available solver, and Mosek [2], an industrial solver with academic licenses.

4.6 BENCHMARK

For comparison of solvers and methods in Section 5.1, Julia features a benchmark package called *BenchmarkTools* [15]. This package executes the tested method multiple times and measures statistics such as elapsed time and memory allocation. The elapsed time is the main factor in our comparison.

All comparisons and evaluations were performed on a MacBook Air with an Apple M1 chip and 16 GB of RAM, using Julia version v1.9.3.

5 RESULTS AND EXPERIMENTS

This chapter covers the results of experiments conducted using the methods outlined in Section 4. Section 5.1 compares methods for entropy maximisation, methods for determining the entropy, and diverse solvers. The results of calculations using real-life data (from Section 3.1) are shown in Section 5.2. The effects of Zhang-Yeung inequalities [21] are evaluated in Section 5.3.

Section 5.4 provides experiments with connected information on data created by randomisation with some fixed underlying conditions and an evaluation of an experiment with data from Section 3.2.

5.1 SOLVERS AND METHODS

Initially, the optimisation methods for directly computing a probability distribution with maximised entropy were implemented as described in Section 4.1. Unfortunately, we were unable to solve the task using exponential cone programming (Subsection 4.1.1) because the Pajarito solver [5] was unable to complete it. Even though the solver can apply both types of constraints (NLP and SOCP), it cannot solve the model with both constraints simultaneously.

The second method for entropy maximisation (Subsection 4.1.2) always returned the same results regardless of the size of fixed marginals for data from Section 3.2. This phenomenon happened with both solvers Ipopt [20] and MadNLP [19], and it is presumably caused by the non-convex nature of the entropic constraints.

The methods that utilised the polymatroid approximation technique consistently delivered valid results when applied to a variety of datasets. Performance comparison of Mosek [2] and SCS [14] solvers and of the methods for determining entropy can be found in the following Subsections 5.1.1 and 5.1.2. Data used for the comparisons are described in Section 3.2.

5.1.1 Solvers Performance Comparison

The test runs varied in the number of dimensions d, ranging from 2 to 10, and the size of fixed marginal entropies m, ranging from 1 to $d-1$. The number of distinct values in each dimension was always $s = 5$.

Table 3 demonstrates that Mosek was consistently faster than SCS during all test runs (the full set of results can be found in Appendix A, Table 14). On the other hand, the time difference was not proportional to the distribution size or the number of fixed marginal entropies. Therefore, the use of the publicly available SCS solver does not significantly impact feasibility due to exceptionally long processing times.

$s = 5$		Time			
		without ZY		with ZY	
		Mosek	SCS	Mosek	SCS
$d = 2$	$m = 1$	0.91 ms	1.7 ms	—	—
$d = 4$	$m = 1$	1.4 ms	7.6 ms	1.4 ms	7.9 ms
	$m = 2$	1.4 ms	28 ms	1.5 ms	18 ms
	$m = 3$	1.2 ms	11 ms	1.3 ms	8.5 ms
$d = 8$	$m = 5$	110 ms	1.1 s	110 ms	910 ms
	$m = 6$	120 ms	650 ms	120 ms	690 ms
	$m = 7$	120 ms	370 ms	120 ms	420 ms
$d = 10$	$m = 5$	12 s	41 s	13 s	31 s
	$m = 6$	17 s	59 s	18 s	56 s
	$m = 7$	19 s	33 s	19 s	36 s
	$m = 8$	22 s	33 s	22 s	34 s
	$m = 9$	21 s	23 s	21 s	23 s

Table 3: Table displaying time needed to compute the maximal entropy for different solvers while using the polymatroid method with estimating the entropy from empirical distribution with d dimension, size of marginal distributions m and a fixed number of samples in each dimension ($s = 5$). Mosek [2] and SCS [14] solvers were used with and without the usage of Zhang-Yeung inequalities (ZY). Data used were from the questionnaire [6] (Section 3.2). The full set of results can be seen in Appendix A in Table 14.

The difference in performance between running the method with Zhang-Yeung inequalities was also negligible.

Even though we had data to try higher dimensions, instances of 11-dimensional distributions with 5 samples or more would require a significant amount of time or greater computing power.

5.1.2 Entropy Methods Performance Comparison

Comparing the methods for determining the entropy revealed notable differences in time requirements. The computation of entropy from the probability distribution using Formula (1) was several times faster (Table 3) than the NSB estimator (Table 4).

Computing the task using a distribution with more than $d = 4$ dimensions was really slow, so it was not used for the comparison. The most time-consuming aspect of maximisation was the calculation of entropy using the Octave implementation [12] of the NSB estimator. The actual solver computation using the NSB estimator was a minor portion of the total time, explaining the marginal differences between the two solvers.

5.1.3 Entropy Methods Accuracy Results

A severe limitation of the method employing the NSB estimator was the precision of entropy computation. The computation occasionally failed when the distribution was not sufficiently undersampled. The Octave code [12] could not determine the distribution's entropy when a precision value

$s = 5$		Time			
		without ZY		with ZY	
		Mosek	SCS	Mosek	SCS
$d = 2$	$m = 1$	3.4 s	3.4 s	—	—
$d = 3$	$m = 1$	5.4 s	5.3 s	—	—
	$m = 2$	9.5 s	9.3 s	—	—
$d = 4$	$m = 1$	7.0 s	6.8 s	6.8 s	6.9 s
	$m = 2$	15.3 s	15.2 s	15.3 s	15.6 s
	$m = 3$	20.7 s	21.4 s	20.7 s	21.4 s

Table 4: Table displaying time needed to compute the maximal entropy for different solvers while using the polymatroid method with estimating the entropy using NSB estimator with d dimensions, size of marginal distributions m and a fixed number of samples in each dimension ($s = 5$). Mosek [2] and SCS [14] solvers were used with and without the usage of Zhang-Yeung inequalities (ZY). Data used were from the questionnaire [6] (Section 3.2).

of 0.1 was required (where 0.1 means approximately a difference of 1/1000 in the resulting entropy).

This issue typically occurred during computations of marginal entropies of size 1 because the size of the marginal distribution was minimal at that point. The minimal size ensured that the distribution was not undersampled. The problem with the number of samples was mitigated by repeatedly rerunning the algorithm with a halved accuracy (doubled precision value) until it returned a valid result (usually lowering the accuracy once was enough).

The smaller accuracy of the result implied that the model of the optimisation task could become infeasible. This problem was resolved by introducing the tolerance t. The resulting constraints were changed from those in Algorithm 3 to pairs of constraints:

$$entropy(p_m) * (1 - t) \leqslant h(m) \leqslant entropy(p_m) * (1 + t) \qquad (16)$$

Inequalities (16) with a tolerance $t = 0.01$ were utilized in all of the solutions, especially during the optimisation of data (Section 5.2).

Sections 5.4.1 and 5.4.2 show the difference in accuracy between the NSB estimator and computation of entropy from the empirical distribution. While the difference in the connected information is negligible for sufficiently sampled distributions, when the distribution is undersampled, the results significantly differ from each other. Those results showed the NSB estimator has significantly better accuracy compared to the empirical distribution method. The results with an undersampled distribution were comparable to those obtained from sufficiently sampled distributions.

Because the NSB estimator is intended for use with undersampled distributions, it failed to compute properly all of the marginal entropies when the distribution was not undersampled. This implies a further limitation on this method because it will fail to determine all of the marginal entropies even with lower precision, therefore lacking sufficient information to run the optimisation task.

Order	Discretised levels							
	2		**3**		**4**		**5**	
	NSB	emp	NSB	emp	NSB	emp	NSB	emp
2	0.727	0.869	0.736	0.745	0.628	0.536	0.500	0.411
3	0.107	0.091	0.094	0.058	0.082	0.038	0.065	0.028
4	0.052	0.019	0.051	0.021	0.047	0.022	0.041	0.023
5	0.031	0.006	0.027	0.012	0.023	0.018	0.018	0.034
6	0.021	0.003	0.021	0.013	0.018	0.043	0.013	0.102
7	0.017	0.002	0.018	0.025	0.016	0.093	0.028	0.181
8	0.017	0.002	0.015	0.041	0.027	0.124	0.096	0.142
9	0.012	0.003	0.014	0.050	0.033	0.091	0.146	0.063
10	0.015	0.003	0.025	0.035	0.126	0.035	0.092	0.016

Table 5: Table showing the normalised connected information I^n/I_N of the fronto-parietal network for different levels of discretisation for both methods of the entropy estimation from data (NSB, empirical distribution - emp) and for all orders. The most significant values are for the order of 2.

5.2 COMPARISON WITH OTHER IMPLEMENTATIONS

The implementation of the maximization algorithms was tested on real data from the article "Network inference and maximum entropy estimation on information diagrams" by Martin et al. [11], and compared to their results, as well as to results mentioned in a bachelor's thesis by Ibatullina [8]. The paper [11] computed connected information on data of resting-state human brain networks.

Connected information from the discretized data were computed. From the partial results, *total correlation* was determined:

Definition 9. Let p be a probability distribution of a random vector $\mathbf{X} = (X_1, \dots, X_n)$ on the sample space χ. The *total correlation* is given by:

$$I_N(\mathbf{X}) = \sum_{i \in N} H(\mathbf{X}_i) - H(\mathbf{X}) = H^1(\mathbf{X}) - H^n(\mathbf{X}).$$

Using Equation (6) for the connected information, the following formula can be constructed:

$$I_N(\mathbf{X}) = \sum_{i=1}^{n} I^k(\mathbf{X})$$

The original paper computed the normalised connected information values of I^2/I_N while discretising into 2 or 3 levels. Ibatullina [8] also computed the connected information for all sizes of marginals and the discretisation into 4 values, but only for the default mode network.

The results are shown in Tables 5 and 6. Throughout all discretisation levels and both methods for entropy estimation, it was evident that the most significant dependence was of order 2. However, the dependence was also higher for the lower levels of discretisation.

The NSB method had more stable results across different discretisation levels (mainly focusing on fixing orders of 2 and 3 as mentioned by Martin

Order	Discretised levels							
	2		**3**		**4**		**5**	
	NSB	emp	NSB	emp	NSB	emp	NSB	emp
2	0.783	0.907	0.811	0.793	0.702	0.589	0.571	0.456
3	0.077	0.064	0.067	0.042	0.055	0.027	0.045	0.020
4	0.054	0.013	0.053	0.013	0.047	0.013	0.040	0.015
5	0.024	0.005	0.022	0.008	0.017	0.016	0.014	0.032
6	0.019	0.002	0.019	0.011	0.016	0.040	0.012	0.100
7	0.012	0.002	0.012	0.023	0.009	0.091	0.019	0.176
8	0.010	0.002	0.009	0.038	0.019	0.111	0.141	0.129
9	0.009	0.003	0.008	0.043	0.120	0.080	0.101	0.056
10	0.011	0.002	0.000	0.029	0.015	0.032	0.056	0.015

Table 6: Table showing the normalised connected information I^n/I_N of the default mode network for different levels of discretisation, both methods of the entropy estimation from data (NSB, empirical distribution - emp) and for all orders. The most significant values are for the order of 2.

Discretisation level	Fronto-parietal network			Default mode network		
	Our results	[11]	Anna	Our results	[11]	Anna
2	0.727	0.89	—	0.783	0.93	0.653
3	0.736	0.94	—	0.811	1.00	0.575
4	0.628	—	—	0.702	—	—
5	0.500	—	—	0.571	—	—

Table 7: Table comparing the result values of I^2/I_N for various levels of discretisation of data from Martin et al. [11] obtained by our research, Martin et al. [11] and Ibatullina [8]. Fields with unknown values are filled with a line.

et al. [11]). While it had lower values for the level 2 of discretisation, the values for levels 4 and 5 were higher than using the estimate from empirical distribution.

Table 7 compares our data with the data computed by Martin et al. [11] and Ibatullina [8]. Despite results from Martin et al. [11] suggesting higher dependence than ours, both sets of results suggested that discretising the data to 3 levels implied higher connected information. This is contrary to the results obtained by Ibatullina [8]. Furthermore, according to the results of Ibatullina [8], some maximal entropies, while fixing two following orders of marginal entropies, were the same, implying exactly zero connected information of some orders (specifically 5, 7 and 9). We argue that having zero connected information across multiple orders, which are not directly consecutive or on the edge of the range of possible orders, is highly unlikely.

5.3 ZHANG-YEUNG INEQUALITIES

We used the Polyhedra package [9] to compute the vertices of an entropic region, ensuring each polymatroid satisfied the entropic constraints for a given order of marginal entropies, both with and without Zhang-Yeung inequalities. We validated the algorithm's correctness by finding the vertex with the maximal value for $h(N)$, the entropy of the entire distribution[1].

However, we were not able to compute the representation for more than four random variables due to the high computation time (using concatenated data from Section 5.2).

Upon comparing the results for fixing the marginal entropies up to orders 2 and 3, we found the vertex representation to be identical regardless of the usage of Zhang-Yeung inequalities. This was different when fixing only the marginal entropies of order 1. While the maximal entropy for the entire distribution remained the same, the vertex representation with Zhang-Yeung inequalities consisted of more vertices, and therefore the entropic regions were not identical.

5.4 EXPERIMENTS AND REAL-LIFE UTILISATION

Differently sized groups of variables interacting with each other can be observed in many areas of everyday life. This section provides insight into the meaning of the connected information in experiments that serve as examples that can be more intuitive and easier to understand. Also, Experiments 5.4.1 and 5.4.2 show the mutual dependence of differently sized groups.

5.4.1 Experiment - Connected Information in Multivariate Normal Distribution

A multivariate normal distribution of n variables is characterized by its covariance matrix Σ of dimension $n \times n$ and its mean vector. However, the mean vector only shifts the origin and does not affect the shape of the distribution or the interactions between the variables. The covariance matrix, which is symmetric and positive-semidefinite, can be expressed as $\Sigma = AA^T$.

The covariance matrix describes the covariance between each pair of variables. In a multivariate normal distribution, we expect to observe interactions of order 2, while interactions among groups of 3 or more should be negligible. These interactions can be analysed using the connected information.

Experiment 1. *Consider a multivariate normal distribution with five variables, defined by a mean vector* **0** *and a covariance matrix* $\Sigma = AA^T$ *where*

$$A = \begin{bmatrix} 0.0 & 0.5 & 1.0 & 1.5 & 2.0 \\ 1.5 & 0.0 & -1.0 & -0.5 & 1.5 \\ 1.0 & -1.0 & 0.5 & 3.0 & -2.5 \\ 0.5 & 1.0 & 2.0 & 0.0 & 1.5 \\ 2.5 & -2.0 & -2.5 & -1.5 & 2.5 \end{bmatrix}.$$

We generated distributions from $10,000,000$ *(and* $1,000$*) random samples* s_i *in each dimension, which were discretised to* x_i *according to Table 8 as follows.*

[1] Each vertex had 2^n coordinates, where n is the number of dimensions of the original distribution, which corresponds to all possible subsets of $N = \{1, \ldots, n\}$

	x_i
$s_i < -10$	1
$-10 \leqslant s_i < -6$	2
$-6 \leqslant s_i < -3$	3
$-3 \leqslant s_i < -1$	4
$-1 \leqslant s_i < 0$	5
$0 \leqslant s_i < 1$	6
$1 \leqslant s_i < 3$	7
$3 \leqslant s_i < 6$	8
$6 \leqslant s_i < 10$	9
$10 \leqslant s_i$	10

Table 8: Table showing the discretisation of random variables in Experiment 1. The first column shows the interval, and the second column shows the discrete index.

Subsequently, we computed the connected information of all orders from both distributions.

Num. of samples	10000000		1000	
Ent. method	NSB	emp	NSB	emp
I^2	1.1672	1.6840	1.0728	1.7804
I^3	0.2065	0.2978	0.1342	0.9503
I^4	0.0103	0.0154	0.0193	1.1503
I^5	0.0020	0.0034	0.0000	0.4251

Table 9: Table showing the connected information of normal distribution measured by different entropy methods (NSB estimator, empirical distribution - emp) on a distribution created from different numbers of samples. The highest connected information is in all cases of order 2.

Num. of samples	10000000		1000	
Ent. method	NSB	emp	NSB	emp
I^2/I^N	0.842	0.842	0.875	0.413
I^3/I^N	0.149	0.149	0.109	0.221
I^4/I^N	0.007	0.008	0.016	0.267
I^5/I^N	0.001	0.002	0.000	0.099

Table 10: Table showing the values I^n/I^N (normalised connected information) of normal distribution measured by different entropy methods (NSB estimator, empirical distribution - emp) on a distribution created from different numbers of samples. The normalised connected information is always the highest for order 2, but when using empirical distribution (emp) with 1000 samples, the difference from other orders is much lower compared to any other column.

Given that the interactions were only pairwise, as indicated by the values in the matrix Σ, the connected information of order 2 was expected to be the highest, with higher-order interactions being nearly zero. Table 9 confirms this expectation, showing that the connected information of order 2 was always the highest. The normalised connected information values in Table 10 are the most intuitive. When there was a sufficient number of samples, both the NSB estimator and the entropy estimation from empirical distribution yielded the same results.

When the distribution was undersampled, the entropy estimation method using empirical distribution yielded inferior results compared to the NSB method. Interestingly, the NSB method performed even better with undersampled distributions than with adequately sampled ones.

5.4.2 Experiment - Connected Information in XOR and RAID 6 Distribution

Another example of connected information involves functions computing complements for data storage. These functions generate a new byte from n original bytes distributed on n disks. The composition of $c + n$ disks can withstand the malfunction or destruction of up to c disks.

XOR is one of the functions that has this ability. Whenever one bit from n original bits is lost, it can be recovered by computing the XOR of the remaining $n - 1$ bits along with the original XOR:

$$x = b_1 \oplus b_2 \oplus \ldots \oplus b_n \Rightarrow \forall i : \; b_i = b_1 \oplus b_2 \oplus \ldots \oplus b_{i-1} \oplus b_{i+1} \oplus \ldots \oplus b_n \oplus x.$$

Another function with the same property is the EVENODD code [3]. It computes the additional byte differently than XOR, and therefore, it is suitable for RAID 6 structures, which are immune to the failure of up to two drives. We applied the EVENODD modification to three 3-bit values (ranging from 0 to 7). Figure 3 illustrates the computation. We can compute the value of any one of b_1, b_2, b_3, provided we know the other two values and the EVENODD result e.

Experiment 2. *Consider* 10^6 *and* 10^3 *random samples from the sample space* $\{0, 1, \ldots, 7\}^3$. *Compute the fourth and fifth value using XOR and EVENODD codes. Then, create a distribution from these samples and compute the connected information of all possible orders using both the NSB method and empirical distribution entropy estimation.*

The results are shown in Tables 11 and 12.

Num. of samples	1,000,000		1,000	
Ent. method	NSB	emp	NSB	emp
I^2	0.0000	0.0001	0.0237	0.1396
I^3	0.0000	0.0007	0.0233	0.7848
I^4	3.3754	4.9995	2.8904	4.4655
I^5	0.1249	0.0000	0.3516	0.0000

Table 11: Table showing the connected information of distribution partially computed by XOR and EVENODD function measured by different entropy methods (NSB estimator, empirical distribution - emp) and on distribution created from different numbers of samples. The highest connected information is always of the order 4.

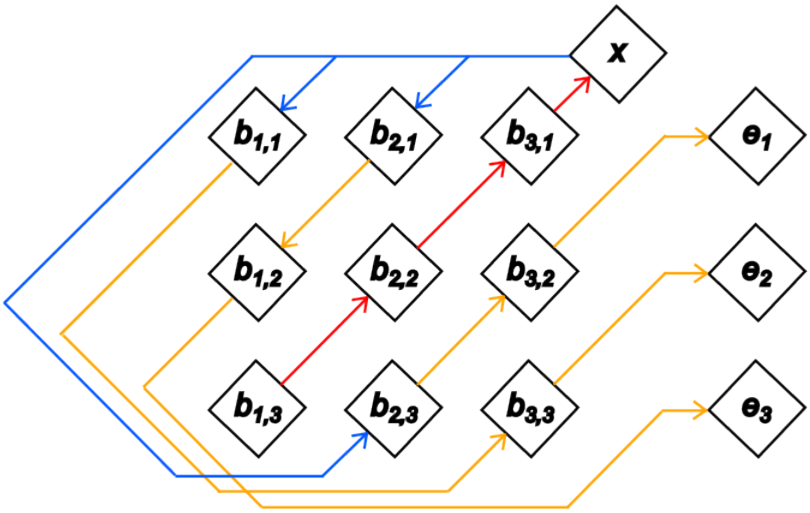

Figure 3: Visualisation of modified EVENODD code that is used in Experiment 2 for the computation of the fifth value. The red line shows the XOR of bits $b_{1,3}, b_{2,2}, b_{3,1}$. The result of this operation is x, which is then used as the first input for the following XOR operations (blue line). The remaining lines (orange) show which other bytes are used in XOR for the resulting bits of the 3-bit number e, which is the result of the whole code.

Num. of samples	1,000,000		1,000	
Ent. method	NSB	Cat.	NSB	Cat.
I^2/I^N	0.000	0.000	0.007	0.026
I^3/I^N	0.000	0.000	0.007	0.146
I^4/I^N	0.964	1.000	0.879	0.828
I^5/I^N	0.036	0.000	0.107	0.000

Table 12: Table showing the values I^n/I^N (normalised connected information) of distribution partially computed by XOR and EVENODD function measured by different entropy methods (NSB estimator, empirical distribution - emp) and on distribution created from different numbers of samples. Normalised connected information is always of the order 4.

In the distribution with enough samples, the connected information of orders 2 and 3 was almost zero. This was due to the random samples from the 3-dimensional space, indicating no interactions that only 2 or 3 variables could describe. A change occurred when we considered 4 variables. The combination of XOR and EVENODD ensured that we could construct the values of the fourth variable from each set of 3 variables, creating a clear dependence in every quadruplet. (It was also possible to determine the fourth or fifth value when we knew the XOR result, EVENODD result, and only 1 variable from the original sample.) There were no interactions of order 5 because both functions have all interactions of order 4, computed from 3 variables.

Both methods for determining entropy computed the highest connected information of order 4 even when the distribution was undersampled. While the empirical distribution method indicated some dependence of order 3, the NSB method successfully computed the lower orders of the connected information as almost zero. However, it did determine the connected information of order 5 to be non-zero. Overall, for undersampled distributions, the NSB method was slightly closer to the results obtained with enough samples.

5.4.3 Experiment - Questionnaire

The last experiment that was carried out was the evaluation of the data described in Section 3.2.

Experiment 3. *Using the questionnaire data from Fokoue and Gunduz [6], compute the connected information both with and without Zhang-Yeung inequalities.*

The results are presented in Table 13.

As expected based on the dataset focusing on potentially connected information, the results revealed a significant dependence within groups of two variables, suggesting strong connections in groups of two. These pairwise connections suggest that knowing any single answer provides a strong basis for predicting the remaining answers. According to Schneidman et al. [16], this finding implies that little additional information is gained

Order	I^n	I^n/I^N
2	10.051	0.828
3	0.8849	0.073
4	0.4212	0.035
5	0.3612	0.030
6	0.2466	0.020
7	0.1289	0.011
8	0.0388	0.003

Table 13: Table showing the connected information and the values I^n/I^N (normalised connected information) of distribution created from student answers to the questionnaire in Example 3. All entropies in the calculations were computed from the empirical distribution. Since the results were the same regardless of Zhang-Yeung inequalities, the table does not specify the usage. The highest connected information is of the order 2.

from observing triplets or larger sets of variables beyond what can be learned from pairs alone.

6 CONCLUSION

In our work, we implemented multiple methods to maximise the information entropy of a distribution. Rewriting the maximisation task into exponential cone optimisation (Subsection 4.1.1) was not successful due to a lack of solvers able to handle both non-linear and exponential cone constraints. Similarly, the direct rewriting of entropic equalities (Subsection 4.1.2) did not succeed because the task was non-convex, preventing solvers from finding the global minimum.

On the other hand, the implementation of the relaxed task using polymatroid and entropic vectors (Section 4.2) proved successful, allowing us to determine the entropy and the connected information of a distribution. We also implemented Zhang-Yeung inequalities [21] and found that these inequalities did not alter the resulting maximal entropy.

For distributions with insufficient samples, we implemented an alternative approach to calculate the information entropy from the data. Instead of constructing an empirical distribution, we used the NSB estimator [13], specifically its implementation in Octave [12]. Comparing the results obtained by both approaches, we observed that the NSB estimator yielded significantly better results for undersampled distributions. However, this improvement was counterweighted by the time required for the computation. We had to pre-compute the values of all entropies to reuse them, thereby minimizing computation time. Implementing the algorithm in Julia instead of Octave could speed up this process by eliminating pipeline calls, making the algorithm suitable for larger datasets without requiring hours of pre-computation.

We applied our algorithms to data from Martin et al. [11] and compared our results with those from the original study and from Ibatullina [8]. Our results are the most accurate, given that the distributions were undersampled and that we used a better entropy estimator. The entropy results from Ibatullina [8] suggested exactly zero connected information across non-consecutive orders.

We showed that, in smaller examples, Zhang-Yeung inequalities [21] do not affect the maximal entropy of a distribution. On the other hand, they can constrain the entropic region, even though they do not affect the maximum entropy.

Finally, calculation over partial data from the paper [6] indicated that the answers to specific questions could be effectively predicted by a single question. This suggests that questions related to the subjective view of a teacher were highly dependent, and having only one question instead of eight could provide a similar amount of information.

In conclusion, this thesis has shown that conventional convex solvers cannot solve the problem of entropy maximisation with entropic constraints. However, it has also shown a promising alternative - the possibility of calculating the connected information of datasets with higher dimensions by relaxing the task. It has obtained the most accurate data when compared with others, and the outcome is the first publicly available method for the computation of maximal entropies and connected information with a focus on undersampled distribution by using the NSB method [12]. Future work could include rewriting the NSB algorithm [12] into Julia, and the most immediate improvements could contain optimisations in data handling of the implemented algorithms.

BIBLIOGRAPHY

[1] Mosek ApS. *Mosek Modeling Cookbook*. https://docs.mosek.com/MOSEKModelingCookbook-letter.pdf(visited 2024-05-19). 2024.

[2] Mosek ApS. *Mosek Optimizer API for Julia. Release 10.1.21*. 2019. URL: https://docs.mosek.com/10.1/juliaapi.pdf.

[3] M. Blaum et al. "EVENODD: an efficient scheme for tolerating double disk failures in RAID architectures." In: *IEEE Transactions on Computers* 44.2 (1995), pp. 192–202. DOI: 10.1109/12.364531.

[4] Qi Chen and Raymond Yeung. "Characterizing the entropy function region via extreme rays." In: Sept. 2012, pp. 272–276. ISBN: 978-1-4673-0224-1. DOI: 10.1109/ITW.2012.6404674.

[5] Chris Coey, Miles Lubin, and Juan Pablo Vielma. "Outer approximation with conic certificates for mixed-integer convex problems." In: *Mathematical Programming Computation* 12.2 (2020), pp. 249–293.

[6] Ernest Fokoue and Necla Gunduz. *Turkiye Student Evaluation*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5S02S. 2013.

[7] David A Huffman. "A method for the construction of minimum-redundancy codes." In: *Proceedings of the IRE* 40.9 (1952), pp. 1098–1101.

[8] Anna Ibatullina. "Entropy maximization under entropic constraints." Czech Technical University in Prague, 2023.

[9] Benoît Legat. "Polyhedral Computation." In: *JuliaCon*. July 2023. URL: https://pretalx.com/juliacon2023/talk/JP3SPX/.

[10] Miles Lubin et al. "JuMP 1.0: Recent improvements to a modeling language for mathematical optimization." In: *Mathematical Programming Computation* (2023). DOI: 10.1007/s12532-023-00239-3.

[11] Elliot Martin et al. "Network Inference and Maximum Entropy Estimation on Information Diagrams." In: *Scientific Reports* 7 (Dec. 2017). DOI: 10.1038/s41598-017-06208-w.

[12] Ilya Nemenman. "Coincidences and Estimation of Entropies of Random Variables with Large Cardinalities." In: *Entropy* 13.12 (2011), pp. 2013–2023. ISSN: 1099-4300. DOI: 10.3390/e13122013. URL: https://www.mdpi.com/1099-4300/13/12/2013.

[13] Ilya Nemenman, Fariel Shafee, and William Bialek. "Entropy and Inference, Revisited." In: *arXiv* 14 (Sept. 2001).

[14] Brendan O'Donoghue et al. "Conic Optimization via Operator Splitting and Homogeneous Self-Dual Embedding." In: *Journal of Optimization Theory and Applications* 169.3 (2016), pp. 1042–1068. URL: http://stanford.edu/~boyd/papers/scs.html.

[15] Jarrett Revels. *BenchmarkTools*. 2015. URL: https://juliaci.github.io/BenchmarkTools.jl/stable/ (visited on 05/06/2024).

[16] Elad Schneidman et al. "Network Information and Connected Correlations." In: *Phys. Rev. Lett.* 91 (23 2003), p. 238701. DOI: 10.1103/PhysRevLett.91.238701. URL: https://link.aps.org/doi/10.1103/PhysRevLett.91.238701.

[17] Thomas Schürmann and Peter Grassberger. "Entropy estimation of symbol sequences." In: *Chaos (Woodbury, N.Y.)* 6 (Oct. 1996), pp. 414–427. DOI: 10.1063/1.166191.

[18] C. E. Shannon. "A Mathematical Theory of Communication." In: *Bell System Technical Journal* 27.3 (1948), pp. 379–423. DOI: https://doi.org/10.1002/j.1538-7305.1948.tb01338.x. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/j.1538-7305.1948.tb01338.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/j.1538-7305.1948.tb01338.x.

[19] Sungho Shin, François Pacaud, and Mihai Anitescu. "Accelerating optimal power flow with GPUs: SIMD abstraction of nonlinear programs and condensed-space interior-point methods." In: *arXiv preprint arXiv:2307.16830* (2023).

[20] Andreas Wächter and Lorenz Biegler. "On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming." In: *Mathematical programming* 106 (Mar. 2006), pp. 25–57. DOI: 10.1007/s10107-004-0559-y.

[21] Zhen Zhang and R.W. Yeung. "On characterization of entropy function via information inequalities." In: *IEEE Transactions on Information Theory* 44.4 (1998), pp. 1440–1452. DOI: 10.1109/18.681320.

Appendices

A BENCHMARK

Table 14: Table displaying the time needed to compute the maximal entropy for different solvers with or without Zhang-Yeung inequalities (ZY) while using the polymatroid method with estimating the entropy from an empirical distribution with d dimension, size of marginal entropies m and a fixed number of samples in each dimension ($s = 5$). Data used were from questionnaire [6] (Section 3.2). SCS solver [10] is slower then Mosek [2], but the ration between them stays similar for all dimensions. There is no significant difference when using the Zhang-Yeung inequalities.

$s = 5$		Time			
		without ZY		with ZY	
		Mosek	SCS	Mosek	SCS
$d = 2$	$m = 1$	0.91 ms	1.7 ms	—	—
$d = 3$	$m = 1$	1.1 ms	6.8 ms	—	—
	$m = 2$	1.0 ms	8.1 ms	—	—
$d = 4$	$m = 1$	1.4 ms	7.6 ms	1.4 ms	7.9 ms
	$m = 2$	1.4 ms	28 ms	1.5 ms	18 ms
	$m = 3$	1.2 ms	11 ms	1.3 ms	8.5 ms
$d = 5$	$m = 1$	2.0 ms	9.9 ms	2.1 ms	11 ms
	$m = 2$	2.3 ms	30 ms	2.3 ms	31 ms
	$m = 3$	1.9 ms	40 ms	2.0 ms	21 ms
	$m = 4$	1.7 ms	12 ms	1.7 ms	15 ms
$d = 6$	$m = 1$	4.3 ms	28.6 ms	4.3 ms	16 ms
	$m = 2$	4.4 ms	210 ms	4.4 ms	150 ms
	$m = 3$	4.3 ms	60 ms	3.9 ms	240 ms
	$m = 4$	4.1 ms	53 ms	4.6 ms	140 ms
	$m = 5$	4.2 ms	28 ms	4.5 ms	45 ms
$d = 7$	$m = 1$	9.7 ms	23 ms	9.7 ms	26 ms
	$m = 2$	11 ms	130 ms	11 ms	130 ms
	$m = 3$	13 ms	420 ms	12 ms	1.1 s
	$m = 4$	14 ms	230 ms	14 ms	440 ms
	$m = 5$	15 ms	350 ms	15 ms	160 ms
	$m = 6$	15 ms	150 ms	15 ms	73 ms
$d = 8$	$m = 1$	30 ms	65 ms	30 ms	64 ms
	$m = 2$	42 ms	720 ms	42 ms	940 ms
	$m = 3$	63 ms	1.5 s	51 ms	1.2 s
	$m = 4$	85 ms	900 ms	85 ms	1.1 s
	$m = 5$	110 ms	1.1 s	110 ms	910 ms
	$m = 6$	120 ms	650 ms	120 ms	690 ms
	$m = 7$	120 ms	370 ms	120 ms	420 ms
$d = 9$	$m = 1$	120 ms	210 ms	120 ms	230 ms
	$m = 2$	180 ms	1.8 s	190 ms	2.1 s

Continued on the next page

Table 14 – continued from the previous page

		Time			
$s = 5$		without ZY		with ZY	
		Mosek	SCS	Mosek	SCS
	$m = 3$	370 ms	4.1 s	370 ms	4.0 s
	$m = 4$	660 ms	3.5 s	650 ms	3.2 s
	$m = 5$	930 ms	4.8 s	1.0 s	5.0 s
	$m = 6$	1.1 s	4.1 s	1.1 s	4.1 s
	$m = 7$	1.3 s	3.4 s	1.3 s	3.5 s
	$m = 8$	1.3 s	2.8 s	1.3 s	2.8 s
$d = 10$	$m = 1$	470 ms	1.2 s	480 ms	1.5 s
	$m = 2$	1.2 s	8.5 s	1.3 s	11 s
	$m = 3$	3.3 s	44 s	3.3 s	38 s
	$m = 4$	7.8 s	19 s	18 s	20 s
	$m = 5$	12 s	41 s	13 s	31 s
	$m = 6$	17 s	59 s	18 s	56 s
	$m = 7$	19 s	33 s	19 s	36 s
	$m = 8$	22 s	33 s	22 s	34 s
	$m = 9$	21 s	23 s	21 s	23 s